

APACHE LUCENE FOR JAVA EE DEVELOPERS

JAVAONE:2015

by [@SanneGrinovero](#)

WHO AM I?

RED HAT

Principal Software Engineer, middleware R&D

Dutch, Italian, living now in London

HIBERNATE TEAM

Hibernate Search project lead

Hibernate ORM

Hibernate OGM

CONTRIBUTING TO INFINISPAN

the Lucene guy: *Infinispan Query*, *Infinispan Lucene Directory*, clustering extensions for *Hibernate Search*

OTHER PROJECTS I HELP WITH...

WildFly, JGroups, *Apache Lucene*, ...

AGENDA

- What is [Apache Lucene](#) and how can it help you
- Integrations with a *JPA application* via [Hibernate Search](#)
- How does this all relate with [Infinispan](#) and [WildFly](#)
- Lucene index management & clouds
- Plans and wishlist for the future

THE SEARCH PROBLEM

*Hey JavaOne, remind me where the talk
having primary key #2342 is?*



All ▼



SQL CAN HANDLE TEXT

The LIKE operator?

LET'S REFRESH SOME LESSONS ON THE WIKIPEDIA

```
Select * from WikipediaPages p where p.content LIKE ?;
```

```
Select * from WikipediaPages p where p.title LIKE ?;
```

```
Select * from WikipediaPages p where  
  (lowercase(p.content) LIKE %:1% OR  
   lowercase(p.content) LIKE %:2% OR  
   lowercase(p.content) LIKE %:3% OR  
   ...);
```



All ▼



A long, empty rectangular search input field with a thin blue border. On the right side of the field, there is a small, grey microphone icon, indicating a voice search function.

Google Search

I'm Feeling Lucky



#lucene|



What's happening?

Gunnar Morling @gunnarmorling
Ha, just submitted pull request
book "1001 tales of ORM pull request"



Yonik Seeley @lucene_solr



Solr/Lucene @SolrLucene



Lucene @lucene



Lucene/Solr Rev @LuceneSolrRev

HOW BAD IS IT?

I'm quoting successfull web companies.

How many can you list which do not provide an effective
search engine?

Human interaction

REQUIREMENTS FOR A SEARCH ENGINE

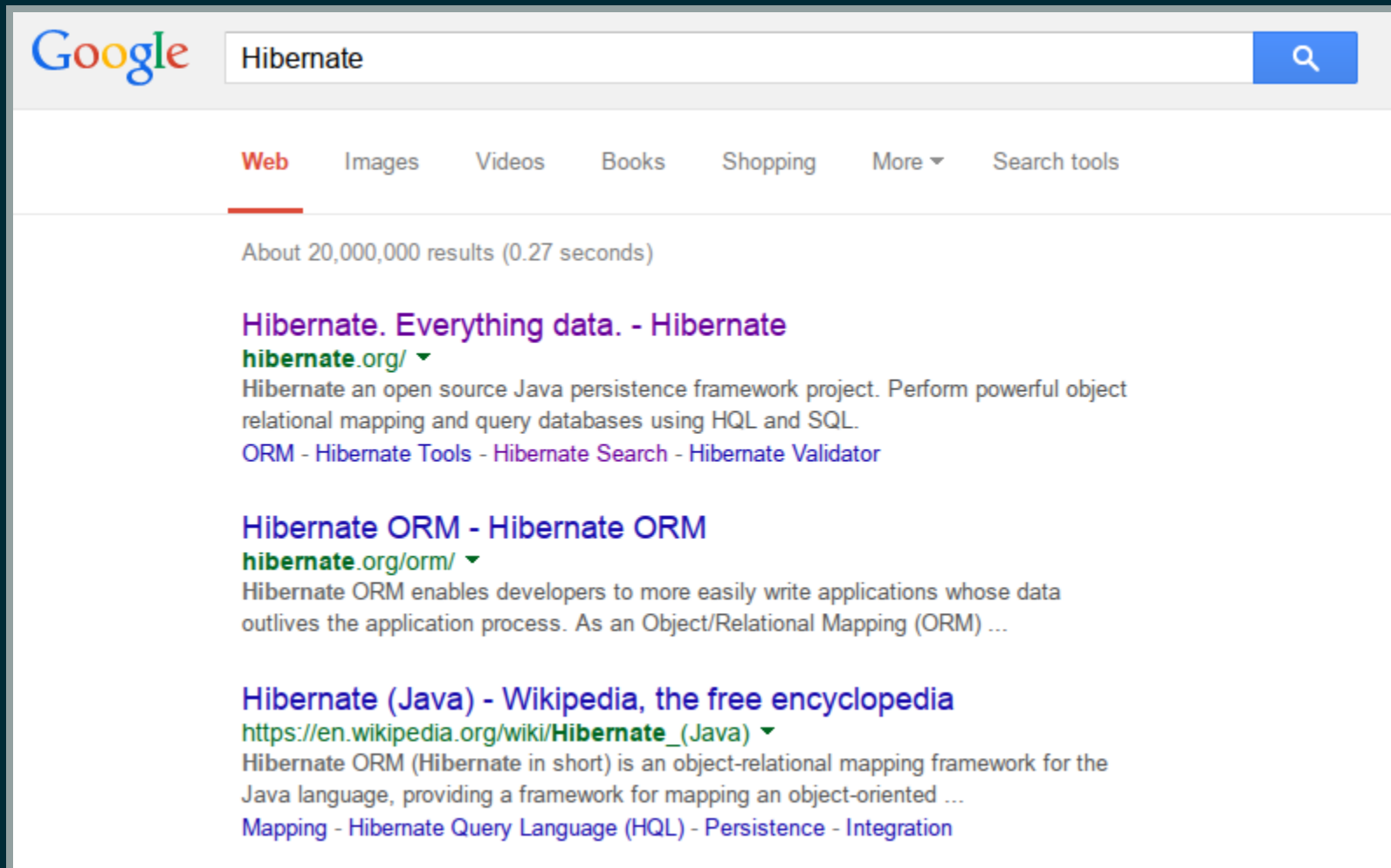
Need to guess what you want w/o you typing all of the content

Humans hate complex forms: can't you just guess what I need?

We want the results in the blink of an eye

We want the right result on top: *Relevance*

we want the right result on top. *Relevance*



The image is a screenshot of a Google search interface. At the top left is the Google logo. To its right is a search bar containing the text 'Hibernate'. A blue search button with a magnifying glass icon is to the right of the search bar. Below the search bar is a horizontal menu with tabs: 'Web' (highlighted with a red underline), 'Images', 'Videos', 'Books', 'Shopping', 'More' (with a dropdown arrow), and 'Search tools'. Below the menu, the search results are displayed. The first result is 'Hibernate. Everything data. - Hibernate' with the URL 'hibernate.org/' and a description: 'Hibernate an open source Java persistence framework project. Perform powerful object relational mapping and query databases using HQL and SQL. ORM - Hibernate Tools - Hibernate Search - Hibernate Validator'. The second result is 'Hibernate ORM - Hibernate ORM' with the URL 'hibernate.org/orm/' and a description: 'Hibernate ORM enables developers to more easily write applications whose data outlives the application process. As an Object/Relational Mapping (ORM) ...'. The third result is 'Hibernate (Java) - Wikipedia, the free encyclopedia' with the URL 'https://en.wikipedia.org/wiki/Hibernate_(Java)' and a description: 'Hibernate ORM (Hibernate in short) is an object-relational mapping framework for the Java language, providing a framework for mapping an object-oriented ... Mapping - Hibernate Query Language (HQL) - Persistence - Integration'.

Google

Hibernate

Web Images Videos Books Shopping More Search tools

About 20,000,000 results (0.27 seconds)

Hibernate. Everything data. - Hibernate
hibernate.org/ ▾
Hibernate an open source Java persistence framework project. Perform powerful object relational mapping and query databases using HQL and SQL.
[ORM](#) - [Hibernate Tools](#) - [Hibernate Search](#) - [Hibernate Validator](#)

Hibernate ORM - Hibernate ORM
hibernate.org/orm/ ▾
Hibernate ORM enables developers to more easily write applications whose data outlives the application process. As an Object/Relational Mapping (ORM) ...

Hibernate (Java) - Wikipedia, the free encyclopedia
[https://en.wikipedia.org/wiki/Hibernate_\(Java\)](https://en.wikipedia.org/wiki/Hibernate_(Java)) ▾
Hibernate ORM (Hibernate in short) is an object-relational mapping framework for the Java language, providing a framework for mapping an object-oriented ...
[Mapping](#) - [Hibernate Query Language \(HQL\)](#) - [Persistence](#) - [Integration](#)

SOME MORE THINGS TO CONSIDER:

Approximate word matches

Stemming / Language specific analysis

Typos

Synonyms, Abbreviations, Technical Language
specializations

BASICS: KEYWORD EXTRACTION

On how to improve running by Scott

1. Tokenization & Analysis:

- how
- improv
- run
- scott

2. Scoring

APACHE LUCENE

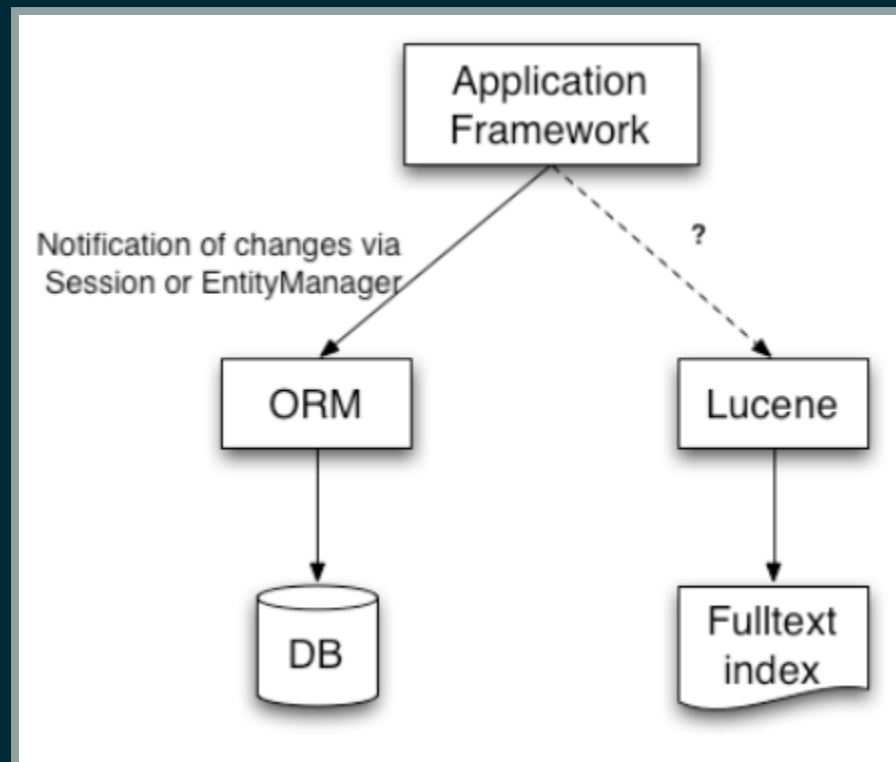
- Open source Apache™ top level project
- Primarily Java, ported to many other languages and platforms
- Extremely popular, it's everywhere!
- High pace of improvement, excellent team
- Most impressive testing

AS A JAVAE DEVELOPER:

You are familiar with the *Java Persistence API*

But Lucene is much better than a relational database to
address this problem

Platform integration?



APACHE LUCENE FROM A JPA PERSPECTIVE



Hibernate Search: deeply but transparently integrated

with Hibernate's *EntityManager*

- Internally uses advanced Apache Lucene features, but protects you from the lower level details
- Gets great performance out of it
- Simple annotations, yet many flexible override options

Does *not* prevent you to perform any form of advanced /
native Lucene query

- Transparent index state synchronization
- Transaction integrations
- Options to rebuild the index efficiently
- Failover and clustering integration points
- Flexible Error handling

HIBERNATE SEARCH QUICKSTART

```
<dependency>
  <groupid>org.hibernate</groupid>
  <artifactid>hibernate-search-orm</artifactid>
  <version>5.5.0.Final</version>
</dependency>
<dependency>
  <groupid>org.hibernate</groupid>
  <artifactid>hibernate-entitymanager</artifactid>
  <version>5.0.3.Final</version>
</dependency>
```

HOW TO INDEX A TRIVIAL DOMAIN MODEL

```
@Entity
public class Actor {

    @Id
    Integer id;

    String name;

}
```

```
@Indexed @Entity  
public class Actor {  
  
    @Id  
    Integer id;  
  
    String name;  
  
}
```

```
@Indexed @Entity  
public class Actor {  
  
    @Id  
    Integer id;  
  
    @Field  
    String name;  
  
}
```

LET'S INTRODUCE RELATIONS

```
@Entity
public class Movie {

    @Id
    Integer id;

    String title;

    @ManyToMany
    Set<Actor> actors = new HashSet<>();

}
```

```
@Indexed @Entity
public class Movie {

    @Id
    Integer id;

    @Field
    String title;

    @ManyToMany @IndexedEmbedded
    Set<Actor> actors = new HashSet<>();

}
```

INDEX FIELDS FOR ACTOR

id	name
1	Harrison Ford
2	Kirsten Dunst

INDEX FIELDS FOR MOVIE

id	title	actors.name *
1	Melancholia	{Kirsten Dunst, Charlotte Gainsbourg, Kiefer Sutherland}
2	The Force Awakens	{Harrison Ford, Mark Hamill, Carrie Fisher}

RUN A LUCENE QUERY & GET JPA MANAGED RESULTS

```
String[] productFields = { "title", "actors.name" };  
  
org.apache.lucene.search.Query luceneQuery = // ...  
  
FullTextEntityManager ftEm =  
    Search.getFullTextEntityManager( entityManager );  
  
FullTextQuery query = // extends javax.persistence.Query  
    ftEm.createFullTextQuery( luceneQuery, Movie.class );  
  
List movies = query.setMaxResults(100).getResultList();  
// Managed entities!  
  
int totalNbrOfResults = query.getResultSize();
```

**LET'S SEE SOME CODE IN
ACTION...**

LUCENE & TEXT ANALYSIS

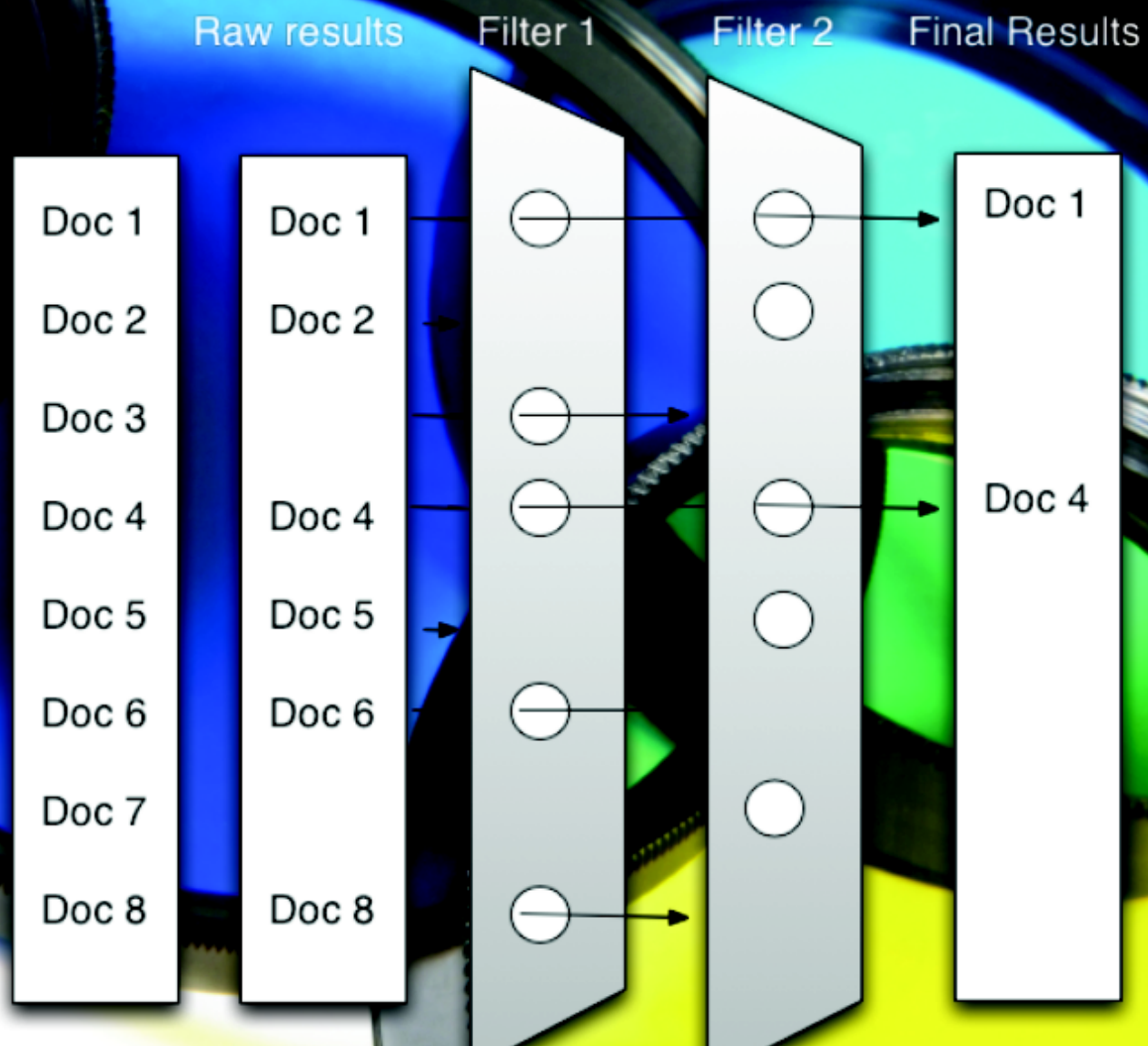
```
@Indexed(index = "tweets")
@Analyzer(definition = "english")
@AnalyzerDef(name = "english",
    tokenizer = @TokenizerDef(
        factory = StandardTokenizerFactory.class),
    filters = {
        @TokenFilterDef(factory = ASCIIFoldingFilterFactory.class),
        @TokenFilterDef(factory = LowerCaseFilterFactory.class),
        @TokenFilterDef(factory = StopFilterFactory.class, params = {
            @Parameter(name = "words", value = "stoplist.properties"),
            @Parameter(name = "ignoreCase", value = "false")
        })
    })
})
@Entity
public class Tweet {
```

FILTERS

```
List results = fullTextEntityManager
    .createFullTextQuery( query, Product.class )
    .enableFullTextFilter( "minorsFilter" )
    .list();
```

```
List results = fullTextEntityManager
    .createFullTextQuery( query, Product.class )
    .enableFullTextFilter( "minorsFilter" )
    .enableFullTextFilter( "specialDayOffers" )
        .setParameter( "day", "20151029" )
    .enableFullTextFilter( "inStockAt" )
        .setParameter( "location", "San Francisco" )
    .list();
```


Filters



FACETING

[Shop All Departments](#) 

Search  [Hibernate Search](#)

[Books](#) [Advanced Search](#) [Browse Subjects](#) [New Releases](#) [Bestsellers](#) [Top](#)

Department
[< Any Department](#)
[< Books](#)
Computers & Internet
[Programming \(14\)](#)
[Computer Science \(4\)](#)
[Databases \(2\)](#)
[Software \(2\)](#)
[Web Development \(2\)](#)
[Networking \(1\)](#)
[Home Computing \(1\)](#)

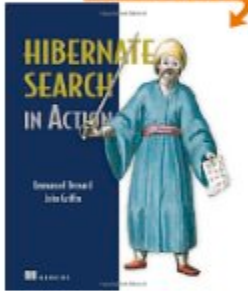
Format
☐ [Paperback \(15\)](#)

Author
Any Author
[Joe Vitale \(1\)](#)

Shipping Option [\(What's this?\)](#)
Any Shipping Option
[Free Super Saver Shipping](#)

Books > Computers & Internet > "Hibernate Search"
Showing 1 - 12 of 15 Results

1.



Hibernate Search in Action I
★★★★★ [\(3 customer reviews\)](#)
Formats
Paperback
Order in the next **2 hours** to get it by **Monday, Apr 18.** ~~\$49.95~~
Only 1 left in stock - order soon.
Eligible for **FREE** Super Saver Shipping.
Excerpt - Page 1: "... breaking the sus
Surprise me! [See a random page in th](#)

2.

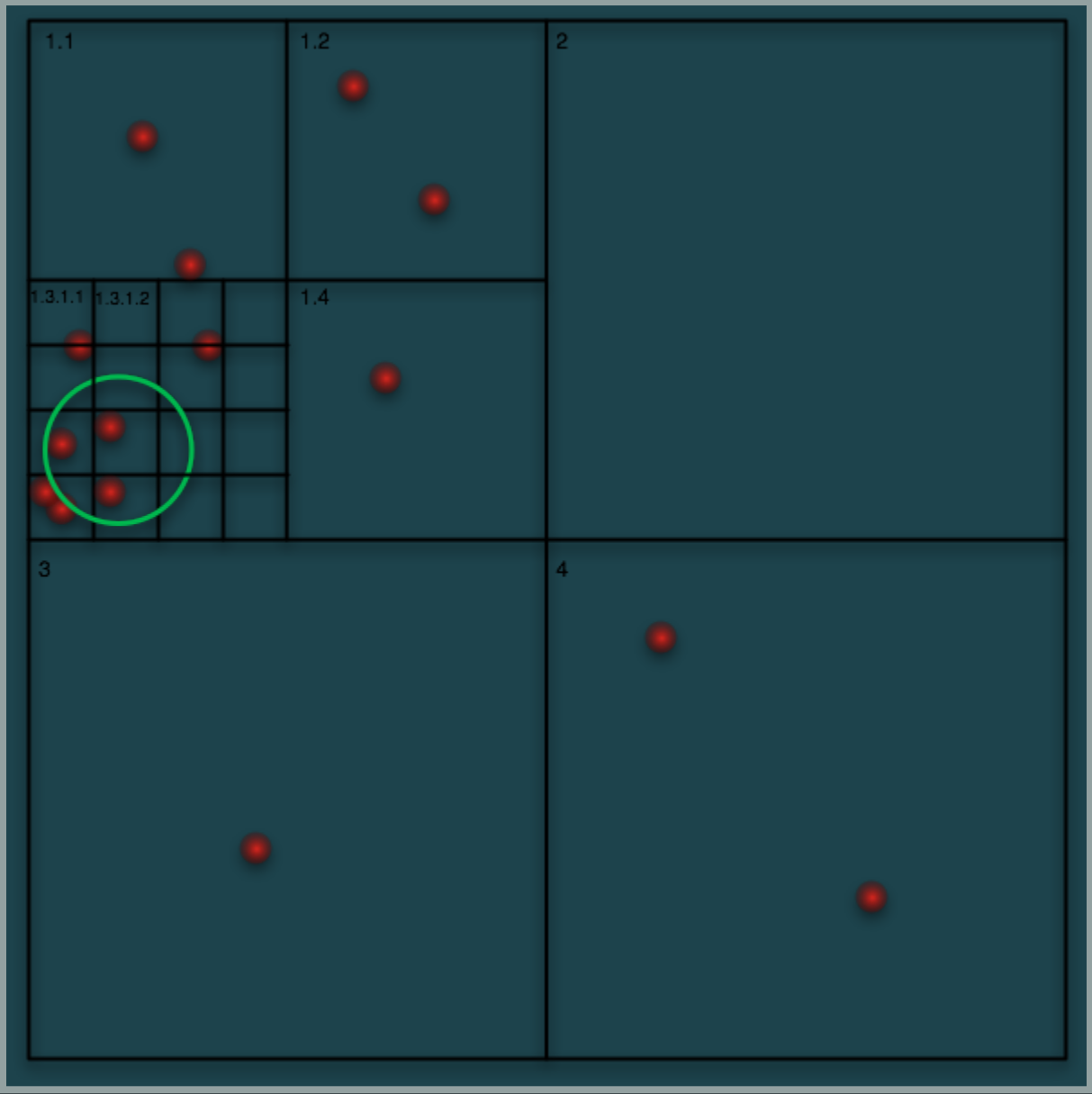


Spring Persistence with Hib
(Nov 2, 2010)
★★★★☆ [\(5 customer reviews\)](#)
Formats
Paperback

"MORE LIKE THIS"

```
Coffee decaffInstance = ... // you already have one
QueryBuilder qb = getCoffeeQueryBuilder();
Query mltQuery = qb
    .moreLikeThis()
    .comparingAllFields()
    .toEntityWithId( decaffInstance.getId() )
    .createQuery();
List results = fullTextEntityManager
    .createFullTextQuery( mltQuery, Coffee.class )
    .list();
```

SPATIAL FILTERING



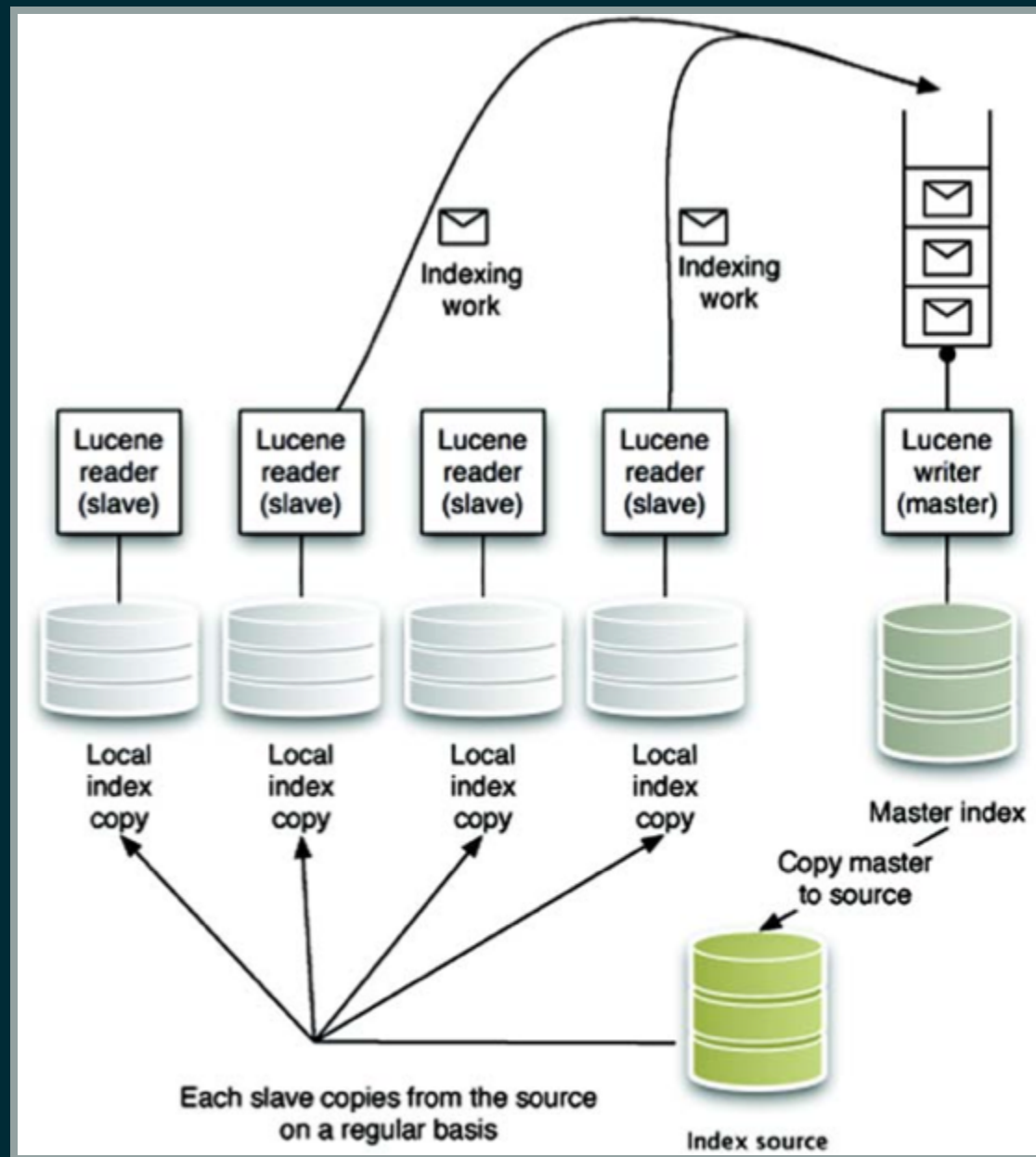
HOW TO RUN THIS ON WILDFLY?

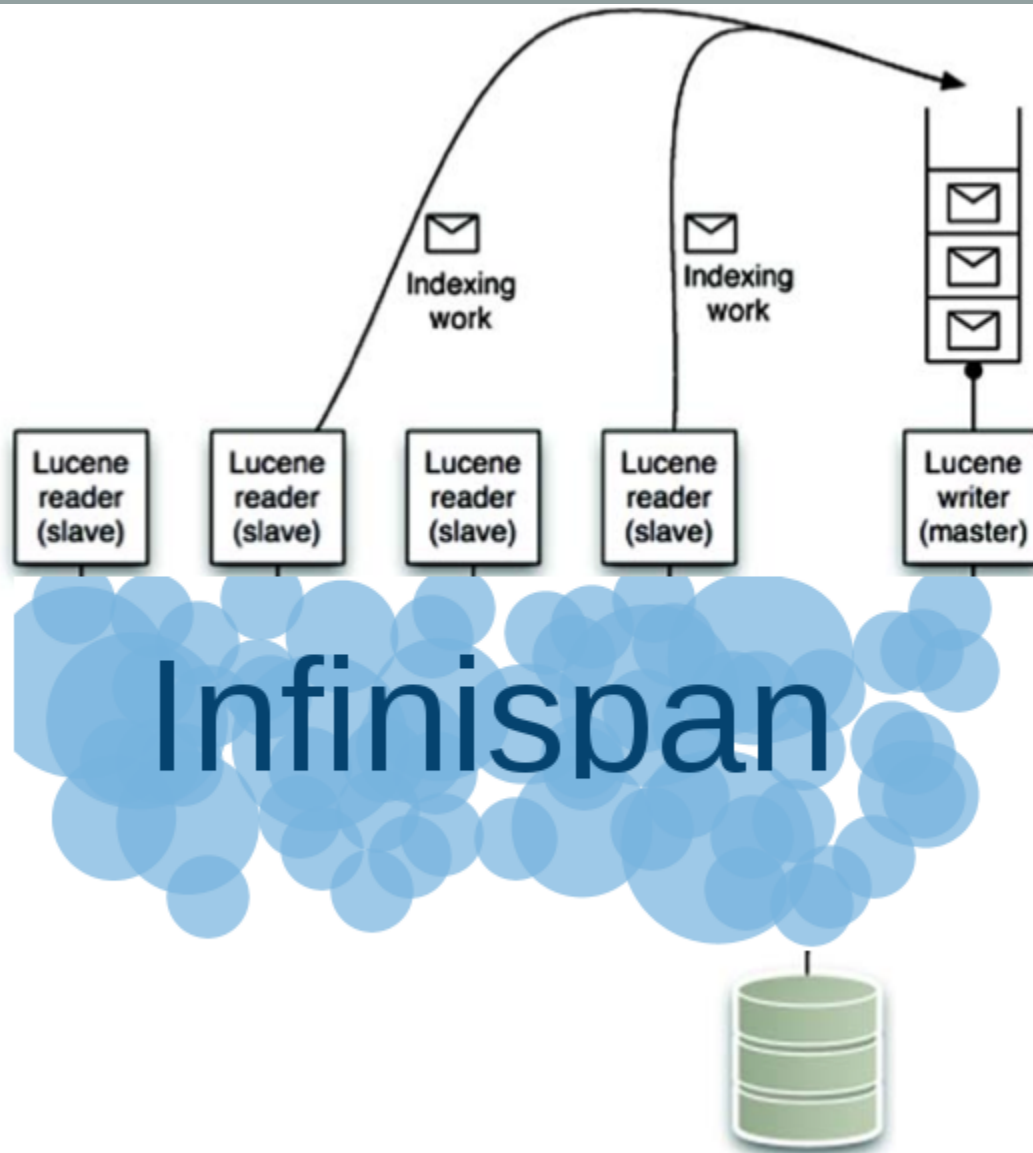
ARCHITECTURE & INDEX MANAGEMENT

ah, the catch!

Indexes need to be stored, updated and read from.

- You can have many indexes, managed independently
- An index can be written by an exclusive writer only
- Backends can be configured differently per index
- Index storage - the *Directory* - can also be configured per index





THE INFINISPAN / LUCENE INTEGRATIONS



WHAT IS INFINISPAN?

- In Memory Key/Value Store
- ASL v2 License
- Scalable
- JTA Transactions
- Persistence (File/JDBC/LevelDB/...)
- Local/Clustered
- Embedded/Server
- ...

INFINISPAN / JAVAEE?

- JavaEE: JCache implementation
- A core component of WildFly / JBoss EAP
- "Embedded" mode does not depend on WildFly
- Hibernate 2n level cache

INFINISPAN / APACHE LUCENE?

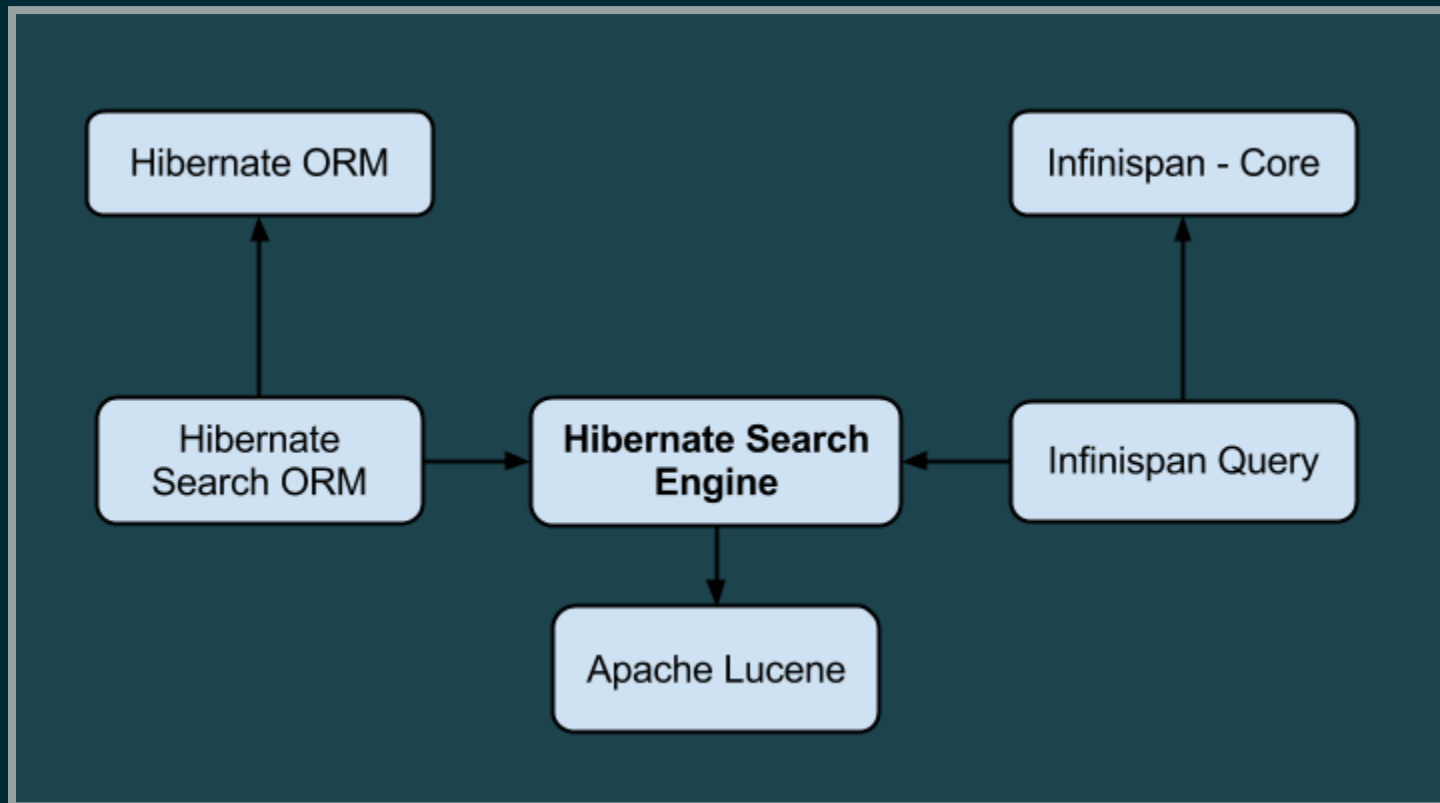
- Lucene integrations for Querying the datagrid!
- Lucene integrations to store the index!
- Hibernate Search integrations!

WHY INFINISPAN?

- You already have it!
- Wildfly, JBoss EAP, and any other Java container supported
- Integrates in any cloud system, esp. OpenShift 3 with Kubernetes and Docker

HIBERNATE SEARCH & INFINISPAN QUERY

- Same underlying technology
- Same API to learn
- Same indexing configuration options



Same annotations, not an entity:

```
@Indexed
@AnalyzerDef(name = "lowercaseKeyword",
    tokenizer = @TokenizerDef
                (factory=KeywordTokenizerFactory.class),
    filters = {
        @TokenFilterDef
            (factory = LowerCaseFilterFactory.class)
    }
)
public class Country {

    @Field(store = Store.YES)
    @Analyzer(definition = "lowercaseKeyword")
    private String name;
```

Store the Java POJO instances in the cache directly:

```
Country uk = ...  
cache.put("UK", uk );
```

USING LUCENE QUERY PARSER

```
QueryParser qp = new QueryParser("default", new StandardAnalyzer(  
  
Query luceneQ = qp  
    .parse("+station.name:airport +year:2014 +month:12 +(avgTemp < 0  
  
CacheQuery cq = Search.getSearchManager(cache)  
    .getQuery(luceneQ, DaySummary.class);  
  
List<Object> results = query.list();
```

COUNT ENTITIES

```
import org.apache.lucene.search.MatchAllDocsQuery;

MatchAllDocsQuery allDocsQuery = new MatchAllDocsQuery();

CacheQuery query = Search.getSearchManager(cache)
                        .getQuery(allDocsQuery, DaySummary.c

int count = query.getResultSize();
```

USING LUCENE INDEXREADER DIRECTLY

```
SearchIntegrator searchFactory = Search.getSearchManager(cache)
    .getSearchFactory();

IndexReader indexReader = searchFactory
    .getIndexReaderAccessor().open(DaySummary.class);

IndexSearcher searcher = new IndexSearcher(indexReader);
```

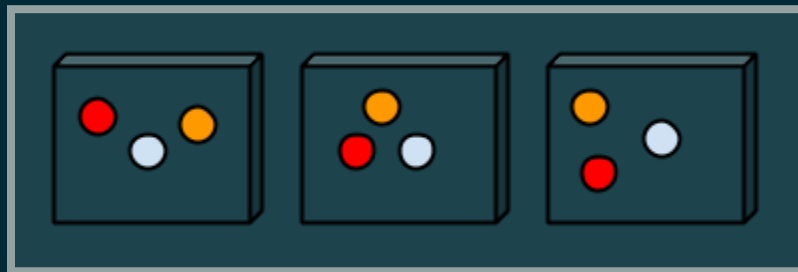
GETTING STARTED WITH INFINISPAN

```
<dependency>  
  <groupId>org.infinispan</groupId>  
  <artifactId>infinispan-embedded</artifactId>  
  <version>8.0.1.Final</version>  
</dependency>
```

```
EmbeddedCacheManager cacheManager = new DefaultCacheManager();  
Cache<String,String> cache = cacheManager.getCache();  
  
cache.put("key", "data goes here");
```

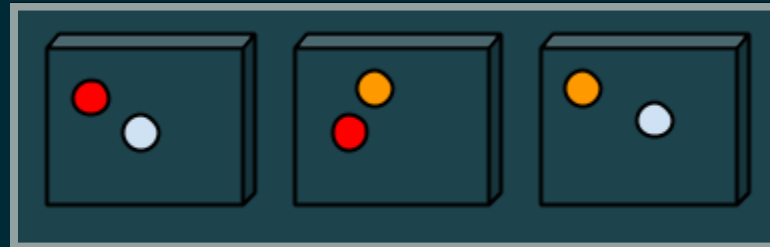
CLUSTERING - REPLICATED

```
Configuration cfg = new ConfigurationBuilder()  
    .clustering().cacheMode(CacheMode.REPL_SYNC)  
    .build();
```



CLUSTERING - DISTRIBUTED

```
Configuration cfg = new ConfigurationBuilder()  
    .clustering().cacheMode(CacheMode.DIST_SYNC)  
    .hash().numOwners(2).numSegments(100)  
    .build();
```



QUERYING

- Apache Lucene Index
- Native Map Reduce
- Index-less
- Hadoop and Spark integration

INDEXING - CONFIGURATION

```
Configuration cfg = new ConfigurationBuilder()  
    .indexing().index(Index.ALL)  
    .build();
```

```
EmbeddedCacheManager cm = new DefaultCacheManager(cfg);  
Cache<Integer, DaySummary> cache = cm.getCache();
```


QUERY - SYNC/ASYNC

```
Configuration cfg = new ConfigurationBuilder()  
    .indexing().index(Index.LOCAL)  
    .addProperty("default.worker.execution", "async")  
    .build();
```

```
EmbeddedCacheManager cm = new DefaultCacheManager(cfg);  
Cache<Integer, DaySummary> cache = cm.getCache();
```

QUERY - RAM STORAGE

```
Configuration cfg = new ConfigurationBuilder()  
    .indexing().index(Index.LOCAL)  
    .addProperty("default.worker.execution", "async")  
    .addProperty("default.directory_provider", "ram")  
    .build();
```

```
EmbeddedCacheManager cm = new DefaultCacheManager(cfg);  
Cache<Integer, DaySummary> cache = cm.getCache();
```

QUERY - INFINISPAN STORAGE

```
Configuration cfg = new ConfigurationBuilder()
    .indexing().index(Index.LOCAL)
    .addProperty("default.worker.execution", "async")
    .addProperty("default.directory_provider", "infinispan")
    .build();

EmbeddedCacheManager cm = new DefaultCacheManager(cfg);
Cache<Integer, DaySummary> cache = cm.getCache();
```

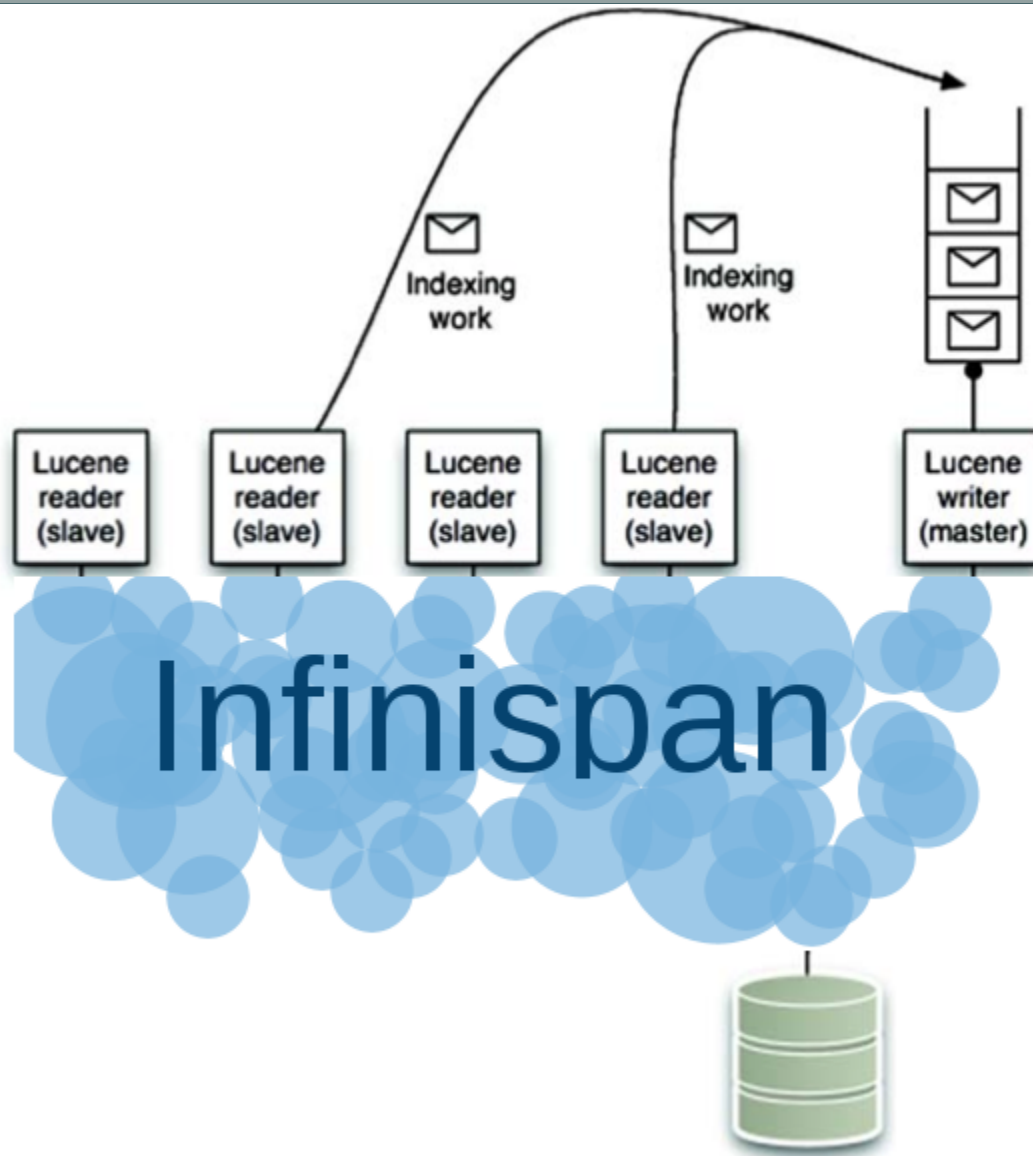
QUERY - FILESYSTEM STORAGE

```
Configuration configuration = new ConfigurationBuilder()  
    .indexing().index(Index.LOCAL)  
        .addProperty("default.directory_provider", "filesystem")  
        .addProperty("default.indexBase", "/path/to/index");  
.build();
```

QUERY - INFINISPAN *INDEXMANAGER*

```
Configuration cfg = new ConfigurationBuilder()
    .indexing().index(Index.LOCAL)
    .addProperty("default.worker.execution", "async")
    .addProperty("default.indexmanager",
        "org.infinispan.query.indexmanager.InfinispanIndexManager")
    .build();

EmbeddedCacheManager cm = new DefaultCacheManager(cfg);
Cache<Integer, DaySummary> cache = cm.getCache();
```



CLOUD OPTIONS RECAP:

- Index replication options
- Index storage options
- Composition and profiles: *IndexManager* or make your own

THE INFINISPAN LUCENE DIRECTORY

Storing the Apache Lucene index in an high-performance
in memory data grid



You don't *need* Hibernate Search to cluster your existing Lucene application, but you'll need some external coordination to guarantee the single IndexWriter.

WHAT'S BEING WORKED ON?

HIBERNATE SEARCH FOR NON-HIBERNATE JPA ?!

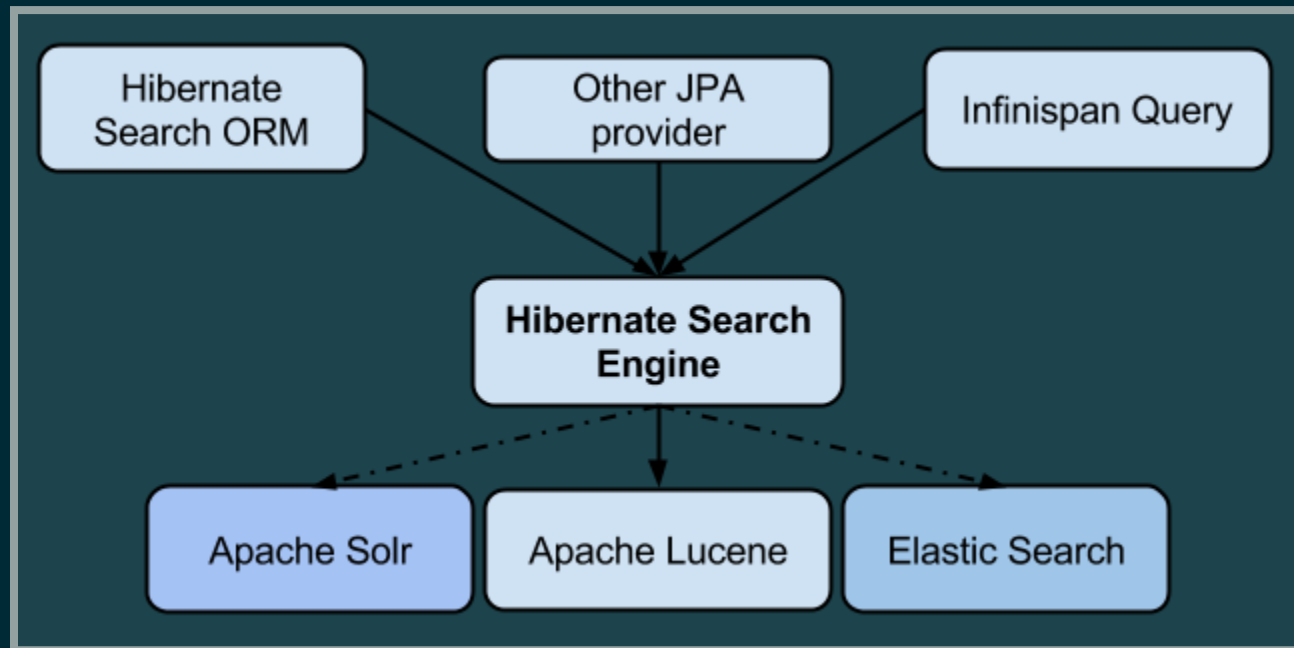
Google Summer of Code student Martin Braun

- Database triggers based indexing bus
(MySQL/MariaDB, PostgreSQL, HSQLDB)
- Works also with *OpenJPA* and *EclipseLink*

github.com/Hotware/Hibernate-Search-GenericJPA

INTEGRATING ELASTICSEARCH AND APACHE SOLR





AND MUCH MORE!

- GSOC student Dmitry Bocharov working on Eclipse tooling
- JBoss Forge 3 integration
- WildFly Swarm fragments
- More performance enhancements for Infinispan indexing
- Example configurations and profiles (HA & JMS master indexing nodes)

See also [the roadmap](#), and help us shape it!

PARTICIPATE!

We love to hear from you

We love contributions

Help yourself and countless others

THANK YOU!

Some references:

- Apache Lucene: lucene.apache.org/core
- My team's blog in.relation.to
- Hibernate Search: hibernate.org/search
 - the super simple Hibernate Search JPA demo:
github.com/Sanne/tweets-cli-demo
 - Source code, with JEE integration tests:
github.com/hibernate/hibernate-search
- WildFly: wildfly.org
- Infinispan: infinispan.org